KaitoroBase: Visual Exploration of Software Architecture Documents

Moon Ting Su, Christian Hirsch, John Hosking
Department of Computer Science
The University of Auckland
Auckland, New Zealand
e-mail: msu010@aucklanduni.ac.nz, chir008@aucklanduni.ac.nz, john@cs.auckland.ac.nz

***Abstract*—This paper describes a software architecture documentation tool (KaitoroBase) built within the Thinkbase Visual Wiki to provide support for non-linear navigation and visualization of Software Architecture Documents (SADs) produced using the Attribute-Driven Design (ADD) method. This involves constructing the meta-model for the SAD in Freebase which provides the foundation for the graph-based interactive visualization enabled by Thinkbase. The resulting tool displays a graphical, high-level structure of SAD, allows for exploratory search, non-linear navigation, and at the same time connects to low-level details of SADs in a wiki.**

Visual Wiki; Software Architecture Document (SAD); Attribute-Driven Design (ADD)

I. INTRODUCTION

"Documenting the architecture is the crowning step to crafting it" [1]. A well-documented architecture facilitates a shorter development time [2]. Software architecture documentation approaches such as IEEE Std 1471-2000 [3] and Views & Beyond (V&B) [4] produce significant amounts of architectural information. This poses difficulties for new stakeholders to locate the right architectural information to perform their tasks. IEEE Std 1471-2000 does not afford effective communication and needs to be extended with guidance to create architecture descriptions which are "accessible" and "understandable" [5]. The need to improve the readability aspect of an architecture document is also evidenced by the use of concept maps to show the main concepts of an architecture document [6]. Furthermore, common approaches to software architecture documentation which use a combination of text-editing tools and modelling tools produce documentations which are typically not used for a number of reasons [7]. This means alternative approaches such as the use of wikis have been explored. Studies have shown that wiki-based Software Architecture Documents (SADs) provide better navigation of the documentation [7]. In a wiki, architecture documents are structured as a set of linked short web pages, with deeper structure. This is good for overview, but poses more difficulty in finding finer details. Another study uses the "Confluence" enterprise wiki as an Architectural Knowledge (AK) management platform [8].

The motivation of this work is to provide support for non-linear navigation and visualization of Software Architecture Documents (SADs). This is done by using a "Visual Wiki". The concept of a Visual Wiki as a knowledge management application combines the notion of a textual and a visual representation. As shown in our previous work [9], this concept can be applied to several domains, using different types of textual repositories (e.g. wikis) and different types of visual representations expressed in a domain specific visual language (DSVL). The main purpose of the Visual Wiki concept is to increase the capability of a text-based knowledge management tool (the wiki), by the synergistic effects of integrating visualizations.

As a proof-of-concept, a software architecture documentation tool (KaitoroBase) is developed "within" the Thinkbase Visual Wiki [10]. Thinkbase is built on top of the Semantic Wiki Freebase [11] and makes use of the Thinkmap visualization framework [12] (Fig. 1).

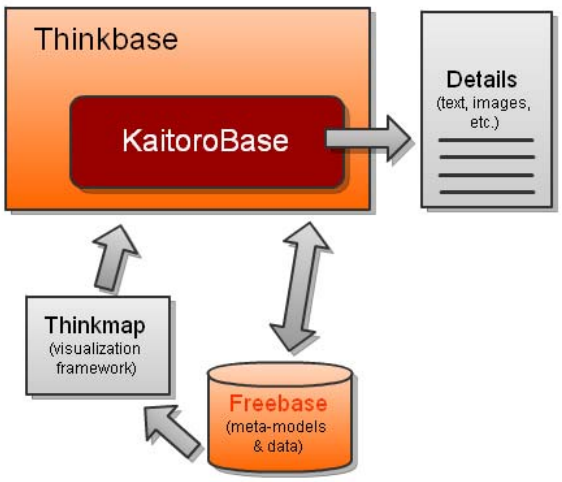

Figure 1. Relationships between the tools

The KaitoroBase is built to capture SADs produced using the Attribute-Driven Design (ADD) method [1]. The related documentation approach is V&B [4]. The KaitoroBase does not impose the use of any particular Architectural Description Language (ADL) to describe the architectural models in the software architecture document. It is a document exploration tool, which focuses specifically on software architecture documents. As such it is different from

1527-1366/09 $29.00 © 2009 IEEE
DOI 10.1109/ASE.2009.26

software architecture design tools such as ArchStudio4.0 [13] and AcmeStudio [14].

The KaitoroBase has the ability to provide a structured high-level view of a SAD and a visual view of the high-level structures. The high-level views are linked to detailed documents (e.g. wiki pages) which can be located anywhere accessible via URLs or attached files such as images. Hence, the high-level views give "context" clues to assist in the navigation of the SAD to locate the needed information.

II. SA Documentation Visual Wiki

The meta-model for SAD in this study was extracted from the ADD method. The meta-model represents the different elements of the SAD and their relationships. The ADD method entails a decomposition approach where the whole system (*module*) is successively decomposed into sub-*modules*. Consequently a SAD contains information about the different *modules* of a system. The decomposition of each *module* at each level is driven by *architectural driver*(s) which are chosen from a set of *concrete quality scenarios* and *functional requirements* pertaining to that decomposition. A suitable *architectural pattern* is constructed based on the *tactics* chosen to fulfil the *concrete quality scenarios*. Then *modules* are instantiated from the *architectural pattern* and represented using different *views* (Module, Component & Connector, and Allocation). *Interfaces* of child *modules* are defined.

The KaitoroBase is built by modelling the SAD meta-model as a Freebase graph data structure (Fig. 2) which provides the foundation for the graph-based interactive visualization enabled by Thinkbase. This is done by creating corresponding "types" in Freebase for all elements in the meta-model and defining their relationships. For example, a "SA3 Module" type is created in Freebase with the properties of level of decomposition, type of module, and so on.

As for the process of creating a specific SAD in Thinkbase, it involves creating instances (known as "topics" in Freebase) of types and connecting them based on the properties of the types. For example, a top-down approach of creating a SAD for a Garage Door System (Fig. 3) can be started by creating a topic called "Garage Door System SAD" of "SA3 Software Architecture Document" type. Its properties ("system name", "company", "version", "system overview" and "module") are then connected to the topics "Garage Door System", "MACNM", "1", "GDS Overview" and "Overall GDS" created from the relevant types.

Thinkbase extracts all the topics including their relationships from Freebase and creates a graph-based interactive visualization. Each topic (e.g. "Overall GDS") is represented as a node in the graph with an icon depending on its type. These icons are used only within the SAD domain. Relationships between topics are represented as typed edges. The graph can be navigated by clicking on the nodes and branches can be expanded and collapsed. The on-demand expansion and shrinking of the nodes allows for an exploratory search and a non-linear graphical navigation of the SAD content. The graph view is shown along side the detailed documentation of the current center node (e.g. attached documents) allowing for a focus-plus-context view.

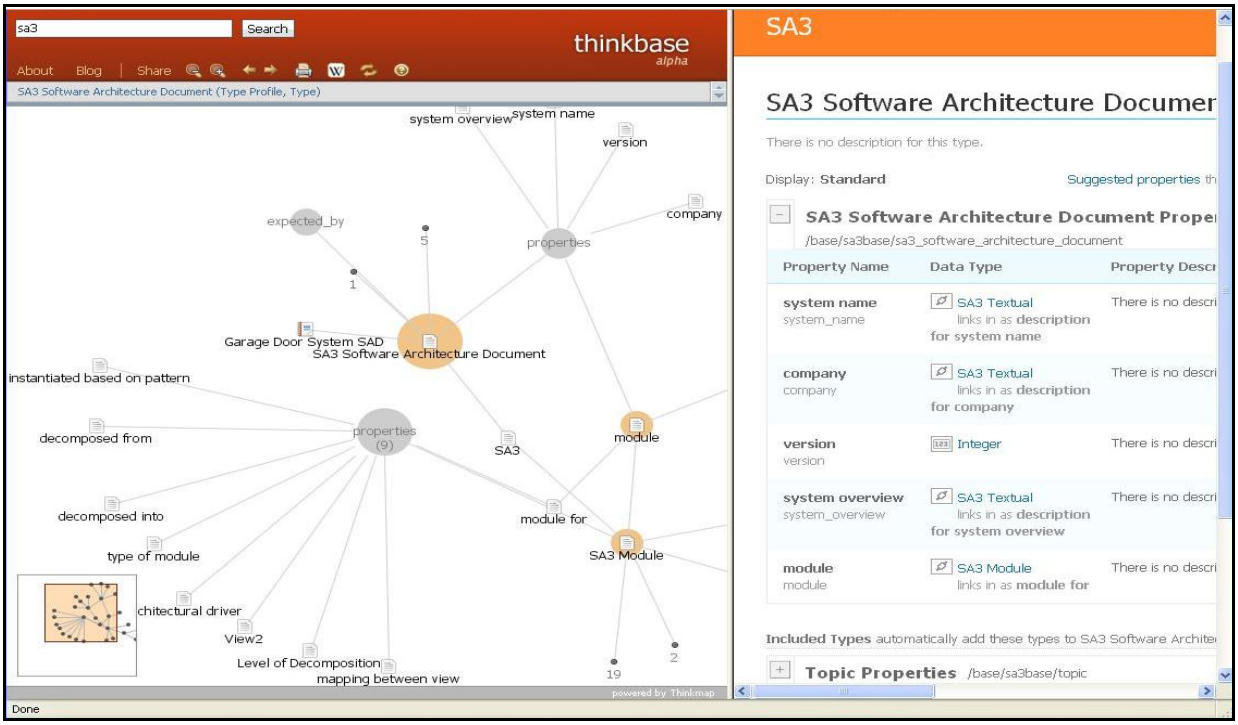

Figure 2. Visual Wiki KaitoroBase tool (meta-model)

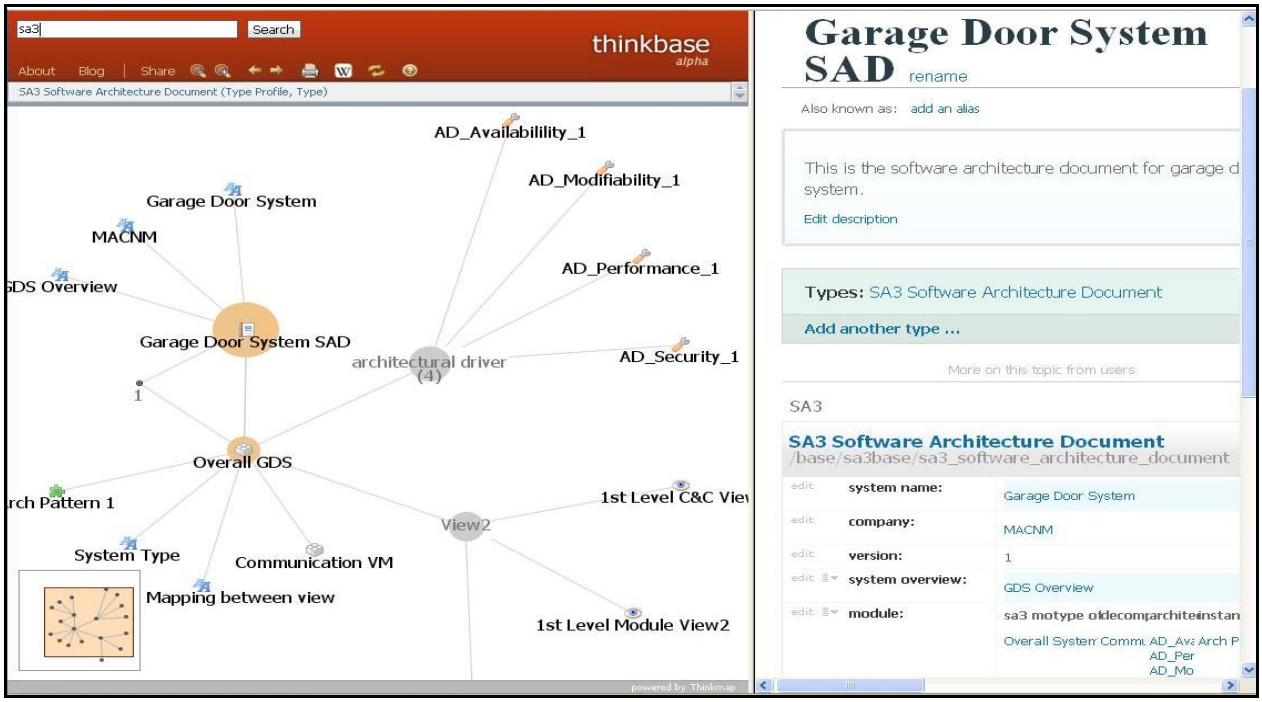

Figure 3. Garage Door System SAD

III. CONCLUSION

This study resulted in a software architecture documentation tool (KaitoroBase) created using the Thinkbase Visual Wiki. In comparison with existing SA documentation approaches, the added advantage of this tool is the graphical visualization of the structure of the entire SAD. The dynamic expansion and collapsing of each element of the SAD and graphically-grouped elements of the same type, reduce the cognitive load on the user. Future work will look into improving the usability of the prototype and evaluating its usefulness. Besides, future work may involve creating parsers to read architectural models described using ADL such as xADL2.0 and Acme into Freebase and display those architectural models in the graph-based interactive visualization enabled by Thinkbase.

To conclude, the tool built in this study provides assistance in non-linear exploration and visualization of SAD by building upon the synergy produced by both the semantic wiki and the visualizations.

ACKNOWLEDGMENT

The authors wish to acknowledge Prof. John Grundy from The University of Auckland for his contribution to this work; the Ministry of Higher Education, Malaysia, FRST and BuildIT, New Zealand and PReSS, University of Auckland for their financial support; and the students taking Software Design and Architecture in Semester 2 2007/2008, from FCSIT, University of Malaya, Malaysia, for the example SA for the Garage Door System.

REFERENCES

[1] L. Bass, P. Clements, and R. Kazman, Software architecture in practice, Addison-Wesley Professional, 2003.

[2] V. Lakshminarayanan, W.Q. Liu, C. L. Chen, S. Easterbrook, and D. E. Perry, Software architects in practice: handling requirements. ACM New York, NY, USA, 2006.

[3] IEEE Std 1471-2000, "Systems and software engineering - Recommended practice for architectural description of software-intensive systems," ISO/IEC 42010 IEEE Std 1471-2000, First edition 15 July 2007, c1-24.

[4] P. Clements, D. Garlan, L. Bass, J. Stafford, R. Nord, J. Ivers, and R. Little, Documenting software architectures: views and beyond, Pearson Education, 2002.

[5] D. Greefhorst, H. Koning, and H. Vliet, "The many faces of architectural descriptions," Information Systems Frontiers, vol. 8, no. 5, 2006, pp.103-113.

[6] H. Koning, and H. Vliet, "Real-life IT architecture design reports and their relation to IEEE Std 1471 stakeholders and concerns, " ASE, vol. 13, no. 2, 2006, pp. 201-223.

[7] F. Bachmann, and P. Merson, "Experience using the web-based tool Wiki for architecture documentation," Technical Note CMU SEI-2005-TN-041, 2005.

[8] R. Farenhorst, and H. Vliet, "Experiences with a Wiki to support architectural knowledge sharing," Proc. of the Wikis4SE @ WikiSym 2008, 2008.

[9] C. Hirsch, J. Hosking, J. Grundy, T. Chaffe, D. Macdonald, and Y. Halytskyy, "The Visual Wiki: A new metaphor for knowledge access and management," Proc. of the 42nd Hawaii International Conference on System Sciences, 2009.

[10] Thinkbase. http://thinkbase.cs.auckland.ac.nz/

[11] Freebase. http://www.freebase.com

[12] Thinkmap. http://www.thinkmap.com

[13] ArchStudio4.0. http://www.isr.uci.edu/projects/archstudio/

[14] AcmeStudio. http://www.cs.cmu.edu/~acme/AcmeStudio/index.html